REMARKS

Status of the Claims

Claims 1 - 8 and 10 - 33 remain pending in the present application. Claim 13 has been amended only to correct a grammatical error.

Claims Rejected Under USC § 103(a)

The Examiner has rejected Claims 1, 3, 15-16, 24-25, 27-28, and 29-31 under 35 USC § 103(a) as being obvious over Corel WordPerfect 6.1, 1996 (hereinafter, "WordPerfect"), in view of Moseberger, D., "The Sane Scanner Interface" (Linux Journal, Issue 47, 3/1998, pp. 1-12, hereinafter "the Sane reference"). Claims 2, 6, 10, 12, and 17 were rejected under 35 USC § 103(a) as being obvious over WordPerfect, in view of the Sane reference, and further in view of "Ulead PhotoImpact 3.0," User Guide for Windows 95 and Windows NT 3.51, 1996, pp. 104-107, 111-114, and 162-167 (hereinafter, "Photoimpact"). Claims 4-5, 18, 21-23, and 32-33 were rejected under 35 USC § 103(a) as being obvious over WordPerfect, in view of the Sane reference, and further in view of U.S. Patent No. 5,907,665 (Sobol et al, hereinafter, "Sobol"). Claims 7-8, 11, and 13-14 were rejected under 35 U.S.C. 103(a) as being upatentable over WordPerfect, in view of the Sane reference, and further in view of one or more of the following: "Mastering Photoshop 5 for the Web," 1998, pp. 1-10; "Troubleshooting and configuring the Windows NT/95 Registry," Clayton Johnson, 1997, pp. 1-2; TWAIN specification version 1.8, 10/22/98; U.S. Patent No. 5,845,076 (Arakawa); the Sobol patent; and U.S. Patent No. 6,154,756 (Hearn). Applicants respectfully disagree that the references disclose all of the steps or elements of the recitation in these claims for at least the reasons set forth below.

In the interest of reducing the complexity of the issues for the Examiner to consider in this response, the following discussion focuses on independent Claims 1, 18, and 24. The patentability of each remaining dependent claim is not necessarily separately addressed in detail. However, applicants' decision not to discuss the differences between the cited art and each dependent claim should not be considered as an admission that applicants concur with the Examiner's conclusion that these dependent claims are not patentable over the disclosure in the cited references. Similarly, applicants' decision not to discuss differences between the prior art and every claim element, or every comment made by the Examiner, should not be considered as an admission that applicants concur with the Examiner's interpretation and assertions regarding those claims. Indeed, applicants believe that all of the dependent claims are patentable over the cited references. Moreover, a specific traverse of the rejection of each

LAW OFFICES OF RONALD M. ANDERSON
600 - 108th Avenue N.E., Suite 507
Bellevue, Washington 98004
Telephone: (425) 688-8816 Fax: (425) 646-6314

dependent claim is not required, since dependent claims are patentable for at least the same reasons as the independent claims from which the dependent claims ultimately depend.

Discussion of the Sane Reference:

SANE is an interface that enables application programs to acquire images from scanning devices through a simple, standardized interface. (See the Sane reference, "Abstract.") Implementation of SANE requires a frontend application and a backend driver. The frontend application, such as a word processor, uses SANE to acquire images from scanning devices. The SANE backend driver directly controls the scanning device using that device's particular communication protocol, insulating the frontend application from device specific details. (See http://www.sane-project.org/man/sane.7.html, "Terminology.") SANE was designed as an alternative to TWAIN, the industry standard scanning device interface, because TWAIN requires that the driver run on the same computer and in the same process as the frontend application. TWAIN also requires the scanner driver contain code for executing a graphical user interface enabled to control the scanner. These requirements make controlling the scanner over a network "unsuitable". (See the Sane reference, "Introduction.") SANE provides a universal interface for operation of image scanning devices that does not require embedding a graphical user interface into the SANE driver, and displaying the graphical user interface in the frontend application. (See the Sane reference, "Introduction.")

However, SANE does not teach or suggest a SANE interface to a scanner implemented by interfacing SANE API calls with a TWAIN driver. Therefore, SANE does not teach or suggest "using a special API module accessed from within an application program for interfacing the application program with a TWAIN module" as recited in Claim 1. Neither SANE nor any of the other cited prior art provides any teaching or suggestion that would lead one of ordinary skill in the art to modify a TWAIN module to provide the functionality of SANE.

Discussion of Implementing SANE using TWAIN

TWAIN is a widely adopted standard; nearly all scanning devices designed for use with a personal computer support TWAIN. SANE, by contrast, is much less common. If someone were able to create a SANE driver that used any TWAIN compliant driver as a backend, the SANE interface would automatically be supported by all TWAIN supported devices. However, no one has implemented SANE by interfacing SANE API calls with a TWAIN driver as required to achieve the SANE functionality, despite the obvious benefits. Moreover, the SANE project creators, authors, and

LAW OFFICES OF RONALD M. ANDERSON
600 - 108th Avenue N.E., Suite 507
Bellevue, Washington 98004
Telephone: (425) 688-8816 Fax: (425) 646-6314

maintainers explicitly teach away from building a SANE driver that utilizes a TWAIN driver as a backend, stating specifically that:

> "If you're familiar with TWAIN, you may wonder why there is a need for *SANE*. Simply put, TWAIN does not separate the user-interface from the driver of a device. This, unfortunately, makes it difficult, if not impossible, to provide network transparent access to image acquisition devices (which is useful if you have a LAN full of machines, but scanners connected to only one or two machines; it's obviously also useful for remote-controlled cameras and such). It also means that any particular TWAIN driver is pretty much married to a particular GUI API (be it Win32 or the Mac API). In contrast, *SANE* cleanly separates device controls from their representation in a user-interface. As a result, *SANE* has no difficulty supporting command-line driven interfaces or network-transparent scanning. **For these reasons, it is unlikely that there will ever be a *SANE backend* that can talk to a TWAIN driver**."(See http://www.sane-project.org/intro.html, last paragraph – emphasis added.)

This statement was last updated on September 23, 2003, indicating that more than five years after the introduction of SANE, and more than four years after the present application was filed, the creators, authors, and maintainers of SANE still believed that building a SANE driver on top of a TWAIN driver was not feasible. Applicants thus find it hard to understand why one of ordinary skill in the art could possibly have been lead to make the combination proposed by the Examiner. If the very authors of one of the cited references teach away from that combination, there is clearly no basis justifying the rejection of applicants' claims as obvious over the combination.

In addition to this statement by authors of the SANE project, all applications built to integrate SANE with TWAIN mentioned on the SANE FAQ Web page (see http://www.sane-project.org/sane-frontends.html) implement a TWAIN driver by interfacing with a SANE driver for a SANE scanner. Specifically, "SaneTwain," "twain-to-sane-bridge," "TWAIN SANE Interface for Mac OS X," "WinSANE," and more, all allow applications to access a SANE scanner through a TWAIN interface by forwarding TWAIN API calls to a SANE driver. There are no applications in the prior art that teach exposing SANE to a user by interfacing a SANE driver with a TWAIN driver. In other words, the combination required to achieve applicants' claimed approach has never been implemented in any applications that employ SANE.

Discussion of WordPerfect

WordPerfect 6.1 is a word processing application that supports integrated TWAIN scanning functionality. WordPerfect directly accesses the TWAIN module. There is no distinct API module used by WordPerfect for interfacing with the TWAIN module. The interface presented to the user is

LAW OFFICES OF RONALD M. ANDERSON
600 - 108th Avenue N.E., Suite 507
Bellevue, Washington 98004
Telephone: (425) 688-8816 Fax: (425) 646-6314

the interface encoded in the TWAIN module, and the user is not isolated from the TWAIN interface by using a special API to provide the user interface presented to the user. Specifically, the reference noted states: "Consult your scanner software documentation for instructions on manipulating the image and ***pasting*** it to WordPerfect." (See WordPerfect 6.1, 1996, "Scan Images Into WordPerfect" – emphasis added.)

WordPerfect 6.1 thus does not teach using a special API module accessed from within an application program for interfacing the application program with a TWAIN module, as required by applicants' claim recitation, and in view of the teaching of the authors of the SANE reference, it would not be obvious to modify TWAIN or WordPerfect to achieve what applicants have recited in their claims.

Discussion of Independent Claim 1

The Examiner has asserted that it would be obvious to one of ordinary skill in the art to combine the teachings of WordPerfect and the Sane reference to separate the TWAIN module from interacting directly with the user. As discussed above, neither the Sane reference nor WordPerfect teach or suggest a special API module accessed from within an application program for interfacing the application program with a TWAIN module. Even assuming, *arguendo*, that there existed a motivation to combine WordPerfect and the Sane reference, the combination of WordPerfect and the Sane reference still does not provided for separating the TWAIN module from interacting directly with the user, because the combination of WordPerfect and the Sane reference does not teach a standardized scanner interface such as SANE implemented by interfacing with existing TWAIN drivers. The Examiner has rejected Claim 1 by combining a prior art reference that teaches a special API scanning module (i.e., SANE) and a prior art reference that teaches integrated TWAIN scanning capability; however, neither prior art reference or the combination of prior art references teach a special API scanning module that interfaces with a TWAIN module. Claim 1 recites: "using a special application programming interface (API) module accessed from within the application program, for interfacing the application program with a TWAIN module that is used for acquiring an image with the image source device that is active, the special API module being entirely separate and distinct from the TWAIN module." Claim 1 thus explicitly recites a special API module for interfacing the application program with a TWAIN module. Therefore, neither WordPerfect, the Sane reference, nor the combination of WordPerfect and SANE teach the elements recited in Claim 1.

LAW OFFICES OF RONALD M. ANDERSON
600 - 108th Avenue N.E., Suite 507
Bellevue, Washington 98004
Telephone: (425) 688-8816 Fax: (425) 646-6314

It would not have been obvious to a person of ordinary skill in the art to combine the teachings of WordPerfect and SANE to separate the TWAIN module from interacting directly with the user. As discussed above, authors of the SANE project explicitly teach away from implementing a SANE backend by interfacing with a TWAIN module. Because the authors of the SANE project teach away from a modification of SANE corresponding to applicants' claimed method, it would not be obvious to a person of ordinary skill in the art to combine the teachings of WordPerfect and SANE, to separate the TWAIN module from interacting directly with the user. Accordingly, the rejection of Claim 1 should be withdrawn.

Because dependent claims are considered to include all of the elements of the independent claims from which the dependent claims ultimately depend and because the cited art does not disclose or suggest all of the elements of independent Claim 1, the rejection of dependent Claims 2-8 and 10-17 under 35 U.S.C. § 103(a) should be withdrawn for at least the same reasons as discussed above in connection with applicants' traverse of the rejection of Claim 1.

Discussion of Independent Claim 18

Claim 18 recites "enabling an image source device user interface provided by a special application programming interface module from within an application program used to create a text content of the document, … wherein the special API (SAPI) module is entirely separate and distinct from a TWAIN module and interacts with the TWAIN module to control the image source device, and wherein the image source device user interface provides a selection scheme that is independent of the TWAIN module within the application program for selecting a plurality of the images stored in the image source device for insertion into the document." The Examiner asserts that the SAPI interface allows the application to direct the acquisition of an image through a "Source Manager" which contacts a "Source Driver," and that this interaction teaches an SAPI module entirely separate and distinct from a TWAIN module. Applicants respectfully submit that both the "Source Manager" and the "Source Driver" are components of the TWAIN module, and therefore cannot be *independent* of the TWAIN module.

Claim 18 recites "wherein the special API module is entirely separate and distinct from a TWAIN module and interacts with the TWAIN module to control the image source device." Claim 18 thus clearly recites an API module "entirely separate and distinct from a TWAIN module;" the presence of multiple components within a particular TWAIN module simply does not teach a second

LAW OFFICES OF RONALD M. ANDERSON
600 - 108th Avenue N.E., Suite 507
Bellevue, Washington 98004
Telephone: (425) 688-8816 Fax: (425) 646-6314

independent module that is capable of interacting with the TWAIN module to control an image source device (e.g., a scanner). In addition, the same argument made in regard to the traverse of Claim 1 also applies in traversing the rejection of Claim 18. Accordingly, the rejection of Claim 18 is unjustified and should be withdrawn.

Because dependent claims are considered to include all of the elements of the independent claims from which the dependent claims ultimately depend and because the cited references do not teach or suggest all of the steps of independent Claim 18, the rejection of dependent Claims 19-23 under 35 U.S.C. § 103(a) should also be withdrawn for at least the same reasons as the rejection of Claim 18.

Discussion of Independent Claim 24

Claim 24 recites "an interface module comprising a special application programming interface (API) module defined by computer-executable instructions stored in the memory, the special API module being entirely separate and distinct from the source manager module and serving as an interface with the source manager module and under control of the application program." In addition, Claim 24 recites some of the same novel and non-obvious elements as recited in Claim 1, for the reasons noted above, and also distinguishes over the cited art for the same reasons as indicated in the traverse of the rejection of Claim 18. Therefore, Claim 24 should be allowed for the same reasons as Claims 1 and 18.

Because dependent claims are considered to include all of the elements of the independent claims from which the dependent claims ultimately depend and because the cited art does not disclose or suggest all of the elements of independent Claim 24, the rejection of dependent Claims 25 - 33 under 35 U.S.C. § 103(a) should be withdrawn for at least the same reasons as the rejection of Claim 24.

Based upon the comments made above, it should be evident that all claims in the present application are patentable. Accordingly, the application should be passed to issue without further delay. If any issues remain, the Examiner is invited to telephone applicants' attorney at the number listed below.

Respectfully submitted,

/ron anderson/
Ronald M. Anderson
Registration No. 28,829

RMA/DWF:elm

LAW OFFICES OF RONALD M. ANDERSON
600 - 108th Avenue N.E., Suite 507
Bellevue, Washington 98004
Telephone: (425) 688-8816 Fax: (425) 646-6314